

1. Расскажите, чем, на ваш взгляд, отличается хорошая верстка от плохой с точки зрения: пользователя; менеджера проекта; дизайнера; серверного программиста.

С точки зрения пользователя, качество верстки сказывается в первую очередь на опыте взаимодействия с интерфейсом или с информацией, которую содержит интерфейс.

Некачественная верстка может испортить впечатление пользователю раз и навсегда, что в свою очередь вредит бизнесу. Это может быть как неудобная ширина текстового контейнера, которая мешает восприятию информации, так и не верно сверстанное поле контактов, которое не открывается по умолчанию в стандартной звонилке на телефоне, или email поле, которое нельзя скопировать или взаимодействовать иным образом. Перечислять можно очень долго, о пользователях нужно думать всегда, на каждом этапе разработке. Не стоит также забывать и о *доступности* интерфейса. Дел там не так много, но почему-то действительно доступные ресурсы встречаются крайне редко.

С точки зрения менеджера проекта... Честно говоря, я никогда не работал с менеджером, поэтому дам волю фантазии.

Менеджер – это связующее звено между клиентом и командой разработчиков, возможно, с его позиции хорошая и плохая верстка сказывается на эффективности командной работы. К примеру, если кто-то сделал работу недобросовестно, это может затормозить дальнейшую работу в ветке написания логики на клиентской стороне или связывание этой самой логики с сервером. В итоге менеджер вынужден двигать сроки, хотя этого можно было избежать.

Например, когда я реализовывал один из своих проектов, стояла задача связать маркеры на карте с карточками объявлений. Объявления должны показываться при клике на маркер.

Я использовал на этапе рендеринга данных с сервера data-атрибуты, а, предположим, если бы мне отдавали готовую верстку и этих атрибутов у меня не было, то процесс связывания маркеров и карточек мог бы затянуться.

С точки зрения дизайнера. Дизайн должен быть функциональным. Хотя я и далек от чувства прекрасного, но пользуясь сервисами и в целом прогуливаясь по сети, нетрудно отличить хороший дизайн от плохого. И вот здесь, многое зависит от самого дизайна. Если он функционален и полезен, а верстальщику лень реализовывать какие-то вещи и их опустили, это одна ситуация. Если наоборот, дизайнер не очень опытный и делает красиво, но не функционально, а по его макетам работает опытный верстальщик, он может изменить что-то в изначальном дизайне или вовсе попросить переделать. В любом случае всегда нужно поддерживать контакт и диалог, чтобы в итоге получился качественный и приятный в использовании интерфейс. Умение найти компромисс в пользу пользовательского опыта это действительно важно в момент взаимодействия дизайнера и разработчика.

С точки зрения серверного программиста. Здесь можно представить ситуацию, когда верстка сделана без проверки на стандартные ситуации.

Допустим, с сервера приходит контент, будь то текст или графика. Он интегрируется в готовую верстку, но верстальщик не проверил свою верстку на переполнение контентом, например. Или не предусмотрел, что в товарные карточки (представим что-то вроде авито, где пользователи

могут загружать свои изображения) люди могут загружать изображения совершенно разных разрешений. И в первом и втором случае верстка разлетится и будет абсолютно некрасивая (дизайнеру будет грустно), не функциональная (фронтенд разработчик тоже грустит) и серверный программист тоже будет грустить (ведь он хотел уйти с работы пораньше, потому что у него голодный котэ дома)...
Всем грустно, а всего лишь надо было предусмотреть пару-тройку простых селекторов в CSS и обсудить с дизайнером, как должен вести себя макет в определенных ситуациях.

2. Опишите основные особенности верстки крупных многостраничных сайтов, дизайн которых может меняться в процессе реализации и поддержки.

Расскажите о своем опыте верстки подобных сайтов: какие методологии, инструменты и технологии вы применяли на практике.

Модульность, методология и тесты.

БЭМ – прекрасно решает задачу модификации, модульности и поддержки крупных проектов. Насколько я знаю, именно для таких целей он и был выпущен в свет.

В своей практике я всегда использую БЭМ методологию (упрощенную) и препроцессор SASS. Они отлично взаимодействуют друг с другом, а если писать селекторы по-человечески, то и для других разработчиков такая структура становится очень приятной и доступной. (less я тоже пробовал и знаю, но в итоге выбрал по логотипу, честно)

Хоть я и не разрабатывал действительно крупные проекты, но переоценить значимость единообразного подхода, модулей и тестирования работы в любом крупном проекте просто невозможно.

3. Опишите основные особенности верстки сайтов, которые должны одинаково хорошо отображаться как на любом современном компьютере, так и на смартфонах и планшетах под управлением iOS и Android. Расскажите о своем опыте верстки подобных сайтов: какие инструменты и технологии вы применяли, как проверяли результат на различных устройствах, какие именно устройства требовалось поддерживать.

В современном мире множество аппаратов и каждый со своим разрешением. В одной ситуации подойдет связка адаптивной + резиновой верстки, в другой – нет.

Я использовал медиа-запросы из CSS3 и, как правило, разработку вел по принципу Mobile-first, также сталкивался с такой вещью как прогрессивное улучшение верстки.

Тестирование проводил с живых устройств под рукой, а также использовал Google DevTools и инструменты разработчика от Mozilla.

Универсальность, на мой взгляд, часто идет в ногу с вредительством, поэтому в идеале нужно четко понимать, кто будущий клиент и как он потенциально будет использовать продукт, чтобы не растрачивать по началу много ресурсов на идеальную подгонку интерфейса для всех устройств.

Для адаптивной графики использовать семантический <picture>, там где можно стараться использовать SVG, на стадии сборки проекта обязательно сжимать всю графику, потому что далеко не везде и всегда есть 4G, а пользователь будет не рад будучи на даче грузить страницу по паре минут.

Про поддерживаемые устройства. Доля мобильных устройств неуклонно растет год к году, поэтому mobile-first подход имеет место быть. Хороший интерфейс везде - это опять же, про хорошую коммуникацию между разработчиком и дизайнером.

4. Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.

В работе я использую плагины для VS Code, которые помогают мне писать код быстрее.

Старый добрый Emmet – это любовь с первого шортката.

Для JavaScript написал клавиатурные сокращения в Alfred и использую их, например: `!jslog` → `console.log()`

Для отладки пользуюсь браузерной консолью.

Ну и конечно же Git. Это не только удобно для логгирования выполненной работы, но еще и спасает время от времени от глупых ошибок и непоправимых ситуаций. Куда ж без этого...

P.S. На самом деле жутко хочется поработать с настоящей продакшен отладкой, посмотреть как работают старшие коллеги с такими задачами, чем пользуются. Я с таких вещей получаю настоящее удовольствие, наблюдая и подмечая, как работают профессионалы.

5. Вам нужно понять, почему страница отображается некорректно в Safari на iOS и в IE на Windows. Код писали не вы, доступа к исходникам у вас нет. Ваши действия? Сталкивались ли вы с подобными проблемами на практике?

На практике сталкивался, но мне повезло, дома есть Windows компьютер и iOS устройство. Но если представить что их нет, то с Windows ситуация решается быстрым развертыванием виртуальной машины (*мне кажется она всегда должна быть, в зависимости от требований по совместимости проекта*), а в случае iOS, на сколько я знаю в XCode имеются средства для тестирования чего угодно. Так или иначе, если быстрого доступа к железе нет, можно справиться виртуальной машиной.

6. Дизайнер отдал вам макет, в котором не показано, как должны выглядеть интерактивные элементы при наведении мыши. Ваши действия?

Если дизайнер в зоне досягаемости, приду к нему. Если нет, пойду к менеджеру проекта. Инициатива в данном случае может быть наказуема. Но если попросят справиться самому, то конечно постараюсь приложить максимум усилий, чтобы выглядело приятно и главное функционально.

**7. Какие ресурсы вы используете для развития в профессиональной сфере? Приведите несколько конкретных примеров (сайты, блоги и так далее).
Какое направление развития вам более близко: JS-программирование, HTML/CSS- верстка или и то, и другое?
Какие ещё области знаний, кроме тех, что непосредственно относятся к работе, вам интересны?**

В почтовом клиенте создал специальную папочку, где собираются email подписки. В настоящий момент читаю рассылки от Dev.to (*нравятся их еженедельные подборки лучших материалов + позволяет тренировать технический Английский*) и HTML Academy (*просто люблю ребят, потому что именно с них в 2014 году начал понимать разницу между плохой версткой и хорошей*), а так же подписан на Habr-ветки по JavaScript, правда далеко не всегда материалы по моему уровню знаний.

Мне однозначно интересно направление в связке HTML+CSS+JS. Потому что так я смогу быть уверен, что интерфейс сделан на совесть, функционален и доступен. Но в настоящий момент хочу максимально углубиться в JavaScript, так как имеется разрыв в знаниях HTML+CSS и JavaScript. (*об этом можно подробнее узнать из моего CV*)

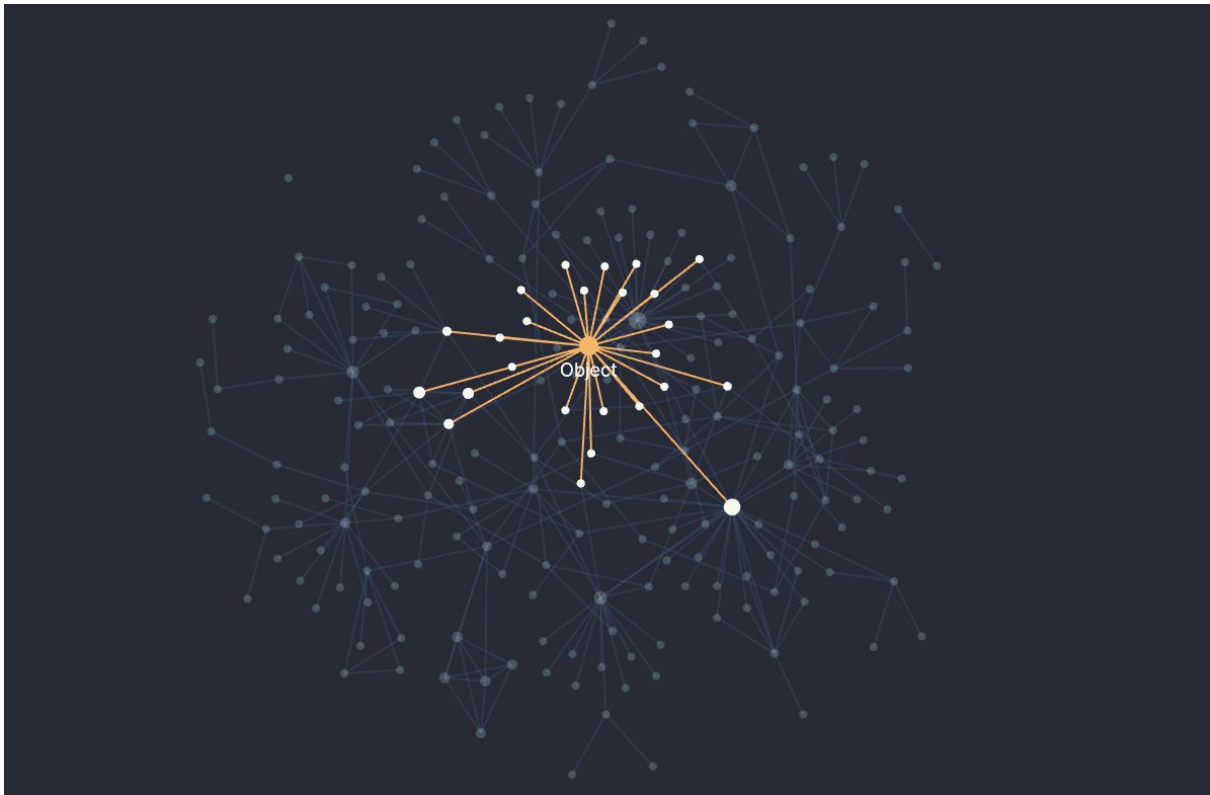
Мне нравится область анализа данных и работы с ними. Это тянется со мной еще со времен прошлой профессиональной деятельности связанной с финансовыми рынками. Обработка информации, построение графиков, диаграмм. Я немного ковырял библиотеку *chart.js* интереса ради, был любопытный опыт. Хотелось бы попробовать в работе и *D3.js*

В целом, мне интересно попробовать поработать буквально со всем. И с Node.js, и с базами данных, посмотреть как

происходит взаимодействие клиента и сервера
максимально в деталях.

Для меня IT не просто хайп. Мне правда все это нравится, но фронтенд нравится в особенности – так как результат трудов чаще всего можно посмотреть и показать.

Например, вот так выглядит моя база знаний, которую я веду для качественного обучения JavaScript.



Это приложение Obsidian для ведения линкованной базы знаний. На сколько я понял, работает на Electron JS. Мне очень нравится эффективность таких заметок и конспектов во время учебы, потому что я могу связать простую информацию с контекстом, а описание кода привязать к теоретической информации.

8. Расскажите нам о себе и предоставьте несколько ссылок на последние работы, выполненные вами.

Для этих целей я создал небольшой сайт, который залил на свой github и использовал github pages в качестве хостинга. Там имеется подробное CV обо мне, выполненные проекты, а также резюме и контакты:

<https://oddman-lab.github.io/>